

A La Carte Embedding: Cheap but Effective Induction of Semantic Feature Vectors

Mikhail Khodak*, Nikunj Saunshi*

Princeton University
{mkhodak, nsaunshi}@princeton.edu

Yingyu Liang

University of Wisconsin-Madison
yliang@cs.wisc.edu

Tengyu Ma

Facebook AI Research
tengyuma@stanford.edu

Brandon Stewart, Sanjeev Arora

Princeton University
{bms4, arora}@princeton.edu

Abstract

Motivations like domain adaptation, transfer learning, and feature learning have fueled interest in inducing embeddings for rare or unseen words, n -grams, synsets, and other textual features. This paper introduces *à la carte* embedding, a simple and general alternative to the usual word2vec-based approaches for building such representations that is based upon recent theoretical results for GloVe-like embeddings. Our method relies mainly on a linear transformation that is efficiently learnable using pretrained word vectors and linear regression. This transform is applicable “on the fly” in the future when a new text feature or rare word is encountered, even if only a single usage example is available. We introduce a new dataset showing how the *à la carte* method requires fewer examples of words in context to learn high-quality embeddings and we obtain state-of-the-art results on a nonce task and some unsupervised document classification tasks.

1 Introduction

Distributional word embeddings, which represent the “meaning” of a word via a low-dimensional vector, have been widely applied by many natural language processing (NLP) pipelines and algorithms (Goldberg, 2016). Following the success of recent neural (Mikolov et al., 2013) and matrix-factorization (Pennington et al., 2014) methods, researchers have sought to extend the approach to other text features, from subword elements to

n -grams to sentences (Bojanowski et al., 2016; Poliak et al., 2017; Kiros et al., 2015). However, the performance of both word embeddings and their extensions is known to degrade in small corpus settings (Adams et al., 2017) or when embedding sparse, low-frequency features (Lazari-dou et al., 2017). Attempts to address these issues often involve task-specific approaches (Rothe and Schütze, 2015; Iacobacci et al., 2015; Pagliarini et al., 2018) or extensively tuning existing architectures such as skip-gram (Poliak et al., 2017; Herbelot and Baroni, 2017).

For computational efficiency it is desirable that methods be able to induce embeddings for only those features (e.g. bigrams or synsets) needed by the downstream task, rather than having to pay a computational *prix fixe* to learn embeddings for all features occurring frequently-enough in a corpus. We propose an alternative, novel solution via *à la carte* embedding, a method which bootstraps existing high-quality word vectors to learn a feature representation in the same semantic space via a linear transformation of the average word embeddings in the feature’s available contexts. This can be seen as a shallow extension of the distributional hypothesis (Harris, 1954), “a feature is characterized by the words in its context,” rather than the computationally more-expensive “a feature is characterized by the features in its context” that has been used implicitly by past work (Rothe and Schütze, 2015; Logeswaran and Lee, 2018).

Despite its elementary formulation, we demonstrate that the *à la carte* method can learn faithful word embeddings from single examples and feature vectors improving performance on important downstream tasks. Furthermore, the approach is resource-efficient, needing only pretrained embed-

dings of common words and the text corpus used to train them, and easy to implement and compute via vector addition and linear regression. After motivating and specifying the method, we illustrate these benefits through several applications:

- **Embeddings of rare words:** we introduce a dataset¹ for few-shot learning of word vectors and achieve state-of-the-art results on the task of representing unseen words using only the definition (Herbelot and Baroni, 2017).
- **Synset embeddings:** we show how the method can be applied to learn more fine-grained lexico-semantic representations and give evidence of its usefulness for standard word-sense disambiguation tasks (Navigli et al., 2013; Moro and Navigli, 2015).
- **n -gram embeddings:** we build seven million n -gram embeddings from large text corpora and use them to construct document embeddings that are competitive with unsupervised deep learning approaches when evaluated on linear text classification.

Our experimental results² clearly demonstrate the advantages of *à la carte* embedding. For word embeddings, the approach is an easy way to get a good vector for a new word from its definition or a few examples in context. For feature embeddings, the method can embed anything that does not need labeling (such as a bigram) or occurs in an annotated corpus (such as a word-sense). Our document embeddings, constructed directly using *à la carte* n -gram vectors, compete well with recent deep neural representations; this provides further evidence that simple methods can outperform modern deep learning on many NLP benchmarks (Arora et al., 2017; Mu and Viswanath, 2018; Arora et al., 2018a,b; Pagliardini et al., 2018).

2 Related Work

Many methods have been proposed for extending word embeddings to semantic feature vectors, with the aim of using them as interpretable and structure-aware building blocks of NLP pipelines (Kiros et al., 2015; Yamada et al., 2016). Many exploit the structure and resources available for specific feature types, such as methods for sense, synsets, and lexemes (Rothe and Schütze, 2015;

Iacobacci et al., 2015) that make heavy use of the graph structure of the Princeton WordNet (PWN) and similar resources (Fellbaum, 1998). By contrast, our work is more general, with incorporation of structure left as an open problem. Embeddings of n -grams are of special interest because they do not need annotation or expert knowledge and can often be effective on downstream tasks. Their computation has been studied both explicitly (Yin and Schütze, 2014; Poliak et al., 2017) and as an implicit part of models for document embeddings (Hill et al., 2016; Pagliardini et al., 2018), which we use for comparison. Supervised and multi-task learning of text embeddings has also been attempted (Wang et al., 2017; Wu et al., 2017).

A main motivation of our work is to learn good embeddings, of both words and features, from only one or a few examples. Efforts in this area can in many cases be split into contextual approaches (Lazaridou et al., 2017; Herbelot and Baroni, 2017) and morphological methods (Luong et al., 2013; Bojanowski et al., 2016; Pado et al., 2016). The current paper provides a more effective formulation for context-based embeddings, which are often simpler to implement, can improve with more context information, and do not require morphological annotation. Subword approaches, on the other hand, are often more compositional and flexible, and we leave the extension of our method to handle subword information to future work. Our work is also related to some methods in domain adaptation and multi-lingual correlation, such as that of Bollegala et al. (2014).

Mathematically, this work builds upon the linear algebraic understanding of modern word embeddings developed by Arora et al. (2018b) via an extension to the latent-variable embedding model of Arora et al. (2016). Although there have been several other applications of this model for natural language representation (Arora et al., 2017; Mu and Viswanath, 2018), ours is the first to provide a general approach for learning semantic features using corpus context.

3 Method Specification

We begin by assuming a large text corpus \mathcal{C}_V consisting of contexts c of words w in a vocabulary \mathcal{V} , with the contexts themselves being sequences of words in \mathcal{V} (e.g. a fixed-size window around the word or feature). We further assume that we have trained word embeddings $\mathbf{v}_w \in \mathbb{R}^d$ on this collo-

¹Dataset: nlp.cs.princeton.edu/CRW

²Code: [www.github.com/NLPPrinceton/ALaCarte](https://github.com/NLPPrinceton/ALaCarte)

cation information using a standard algorithm (e.g. word2vec / GloVe). Our goal is to construct a good embedding $\mathbf{v}_f \in \mathbb{R}^d$ of a text feature f given a set \mathcal{C}_f of contexts it occurs in. Both f and its contexts are assumed to arise via the same process that generates the large corpus \mathcal{C} . In many settings below, the number $|\mathcal{C}_f|$ of contexts available for a feature f of interest is much smaller than the number $|\mathcal{C}_w|$ of contexts that the typical word $w \in \mathcal{V}$ occurs in. This could be because the feature is rare (e.g. unseen words, n -grams) or due to limited human annotation (e.g. word senses, named entities).

3.1 A Linear Approach

A naive first approach to construct feature embeddings using context is *additive*, i.e. taking the average over all contexts of a feature f of the average word vector in each context:

$$\mathbf{v}_f^{\text{additive}} = \frac{1}{|\mathcal{C}_f|} \sum_{c \in \mathcal{C}_f} \frac{1}{|c|} \sum_{w \in c} \mathbf{v}_w \quad (1)$$

This formulation reflects the training of commonly used embeddings, which employs additive composition to represent the context (Mikolov et al., 2013; Pennington et al., 2014). It has proved successful in the bag-of-embeddings approach to sentence representation (Wieting et al., 2016; Arora et al., 2017), which can compete with LSTM representations, and has also been given theoretical justification as the *maximum a posteriori* (MAP) context vector under a generative model related to popular embedding objectives (Arora et al., 2016). Lazaridou et al. (2017) use this approach to learn embeddings of unknown word amalgamations, or *chimeras*, given a few context examples.

The additive approach has some limitations because the set of all word vectors is seen to share a few common directions. Simple addition amplifies the component in these directions, at the expense of less common directions that presumably carry more “signal.” Stop-word removal can help to ameliorate this (Lazaridou et al., 2017; Herbelot and Baroni, 2017), but does not deal with the fact that content-words also have significant components in the same direction as these deleted words. **Another mathematical framework to address this lacuna is to remove the top one or top few principal components**, either from the word embeddings themselves (Mu and Viswanath, 2018) or from their summations (Arora et al., 2017). However, this approach is liable to either not remove

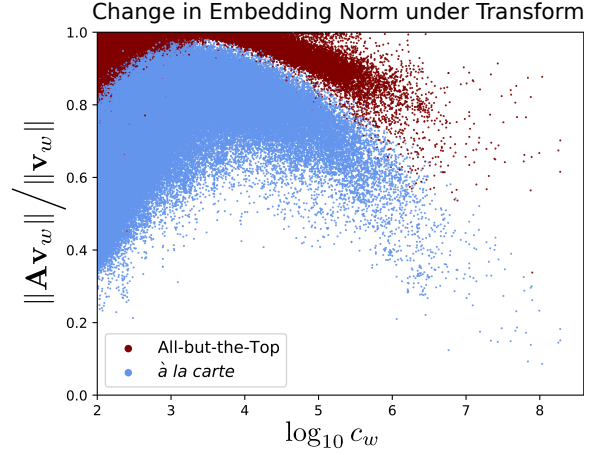


Figure 1: Plot of the ratio of embedding norms after transformation as a function of word count. While All-but-the-Top tends to affect only very frequent words, *à la carte* learns to remove components even from less common words.

enough noise or cause too much information loss without careful tuning (c.f. Figure 1).

We now note that removing the component along the top few principal directions is tantamount to multiplying the additive composition by a fixed (but data-dependent) matrix. Thus a natural extension is to use an arbitrary linear transformation which will be *learned* from the data, and hence guaranteed to do at least as well as any of the above ideas. Specifically, we find the transform that can best recover *existing* word vectors \mathbf{v}_w —which are presumed to be of high quality—from their additive context embeddings $\mathbf{v}_w^{\text{additive}}$. This can be posed as the following linear regression problem

$$\mathbf{v}_w \approx \mathbf{A} \mathbf{v}_w^{\text{additive}} = \mathbf{A} \left(\frac{1}{|\mathcal{C}_w|} \sum_{c \in \mathcal{C}_w} \sum_{w' \in c} \mathbf{v}_{w'} \right) \quad (2)$$

where $\mathbf{A} \in \mathbb{R}^{d \times d}$ is learned and we assume for simplicity that $\frac{1}{|c|}$ is constant (e.g. if c has a fixed window size) and is thus subsumed by the transform. After learning the matrix, we can embed any text feature in the same semantic space as the word embeddings via the following expression:

$$\mathbf{v}_f = \mathbf{A} \mathbf{v}_f^{\text{additive}} = \mathbf{A} \left(\frac{1}{|\mathcal{C}_f|} \sum_{c \in \mathcal{C}_f} \sum_{w \in c} \mathbf{v}_w \right) \quad (3)$$

Note that \mathbf{A} is fixed for a given corpus and set of pretrained word embeddings and so does not need to be re-computed to embed different features or feature types.

Algorithm 1: The basic *à la carte* feature embedding induction method. All contexts c consist of sequences of words drawn from the vocabulary \mathcal{V} .

Data: vocabulary \mathcal{V} , corpus $\mathcal{C}_{\mathcal{V}}$, vectors $\mathbf{v}_w \in \mathbb{R}^d \forall w \in \mathcal{V}$, feature f , corpus \mathcal{C}_f of contexts of f

Result: feature embedding $\mathbf{v}_f \in \mathbb{R}^d$

```

1 for  $w \in \mathcal{V}$  do
2   let  $\mathcal{C}_w \subset \mathcal{C}_{\mathcal{V}}$  be the subcorpus of contexts of  $w$ 
3    $\mathbf{u}_w \leftarrow \frac{1}{|\mathcal{C}_w|} \sum_{c \in \mathcal{C}_w} \sum_{w' \in c} \mathbf{v}_{w'}$  // compute each word's context embedding  $\mathbf{u}_w$ 
4    $\mathbf{A} \leftarrow \arg \min_{\mathbf{A} \in \mathbb{R}^{d \times d}} \sum_{w \in \mathcal{V}} \|\mathbf{v}_w - \mathbf{A} \mathbf{u}_w\|_2^2$  // compute context-to-feature transform  $\mathbf{A}$ 
5    $\mathbf{u}_f \leftarrow \frac{1}{|\mathcal{C}_f|} \sum_{c \in \mathcal{C}_f} \sum_{w \in c} \mathbf{v}_w$  // compute feature's context embedding  $\mathbf{u}_f$ 
6    $\mathbf{v}_f \leftarrow \mathbf{A} \mathbf{u}_f$  // transform feature's context embedding

```

Theoretical Justification: As shown by [Arora et al. \(2018b, Theorem 1\)](#), the approximation (2) holds exactly in expectation for some matrix \mathbf{A} when contexts $c \in \mathcal{C}$ are generated by sampling a context vector $\mathbf{v}_c \in \mathbb{R}^d$ from a zero-mean Gaussian with fixed covariance and drawing $|c|$ words using $\mathbb{P}(w|\mathbf{v}_c) \propto \exp\langle \mathbf{v}_c, \mathbf{v}_w \rangle$. The correctness (again in expectation) of (3) under this model is a direct extension. [Arora et al. \(2018b\)](#) use large text corpora to verify their model assumptions, providing theoretical justification for our approach. We observe that the best linear transform \mathbf{A} can recover vectors with mean cosine similarity as high as 0.9 or more with the embeddings used to learn it, thus also justifying the method empirically.

3.2 Practical Details

The basic *à la carte* method, as motivated in Section 3.1 and specified in Algorithm 1, is straightforward and parameter-free (the dimension d is assumed to have been chosen beforehand, along with the other parameters of the original word embeddings). In practice we may wish to modify the regression step in an attempt to learn a better transformation matrix \mathbf{A} . However, the standard first approach of using ℓ_2 -regularized (Ridge) regression instead of simple linear regression gives little benefit, even when we have more parameters than word embeddings (i.e. when $d^2 > |\mathcal{V}|$).

A more useful modification is to weight each point by some non-decreasing function α of each word's corpus count c_w , i.e. to solve

$$\mathbf{A} = \arg \min_{\mathbf{A} \in \mathbb{R}^{d \times d}} \sum_{w \in \mathcal{V}} \alpha(c_w) \|\mathbf{v}_w - \mathbf{A} \mathbf{u}_w\|_2^2 \quad (4)$$

where \mathbf{u}_w is the additive context embedding. This reflects the fact that more frequent words likely

have better pretrained embeddings. In settings where $|\mathcal{V}|$ is large we find that a hard threshold ($\alpha(c) = \mathbf{1}_{c \geq \tau}$ for some $\tau \geq 1$) is often useful. When we do not have many embeddings we can still give more importance to words with better embeddings via a function such as $\alpha(c) = \log c$, which we use in Section 5.1.

4 One-Shot and Few-Shot Learning of Word Embeddings

While we can use our method to embed any type of text feature, its simplicity and effectiveness is rooted in word-level semantics: the approach assumes pre-existing high quality word embeddings and only considers collocations of features with words rather than with other features. Thus to verify that our approach is reasonable we first check how it performs on word representation tasks, specifically those where word embeddings need to be learned from very few examples. In this section we first investigate how representation quality varies with number of occurrences, as measured by performance on a similarity task that we introduce. We then apply the *à la carte* method to two tasks measuring the ability to learn new or synthetic words from context, achieving strong results on the nonce task of [Herbelot and Baroni \(2017\)](#).

4.1 Similarity Correlation vs. Sample Size

Performance on pairwise word similarity tasks is a standard way to evaluate word embeddings, with success measured via the Spearman correlation between a human score and the cosine similarity between word vectors. An overview of widely used datasets is given by [Faruqui and Dyer \(2014\)](#). However, none of these datasets can be used directly to measure the effect of word frequency on

embedding quality, which would help us understand the data requirements of our approach. We address this issue by introducing the *Contextual Rare Words* (CRW) dataset, a subset of 562 pairs from the Rare Word (RW) dataset (Luong et al., 2013) supplemented by 255 sentences (contexts) for each rare word sampled from the Westbury Wikipedia Corpus (WWC) (Shaoul and Westbury, 2010). In addition we provide a subset of the WWC from which all sentences containing these rare words have been removed. The task is to use embeddings trained on this subcorpus to induce rare word embeddings from the sampled contexts.

More specifically, the CRW dataset is constructed using all pairs from the RW dataset where the rarer word occurs between 512 and 10000 times in WWC; this yields a set of 455 distinct rare words. The lower bound ensures that we have a sufficient number of rare word contexts, while the upper bound ensures that a significant fraction of the sentences from the original WWC remain in the subcorpus we provide. In CRW, the first word in every pair is the more frequent word and occurs in the subcorpus, while the second word occurs in the 255 sampled contexts but not in the subcorpus. We provide word2vec embeddings trained on all words occurring at least 100 times in the WWC subcorpus; these vectors include those assigned to the first (non-rare) words in the evaluation pairs.

Evaluation: For every rare word the method under consideration is given eight disjoint subsets containing 1, 2, 4, ..., 128 example contexts. The method induces an embedding of the rare word for each subset, letting us track how the quality of rare word vectors changes with more examples. We report the Spearman ρ (as described above) at each sample size, averaged over 100 trials obtained by shuffling each rare word’s 255 contexts.

The results in Figure 2 show that our *à la carte* method significantly outperforms the additive baseline (1) and its variants, including stop-word removal, SIF-weighting (Arora et al., 2017), and top principal component removal (Mu and Viswanath, 2018). We find that combining SIF-weighting and top component removal also beats these baselines, but still does worse than our method. These experiments consolidate our intuitions from Section 3 that removing common components and frequent words is important and that learning a data-dependent transformation is an effective way to do this. However, if we train

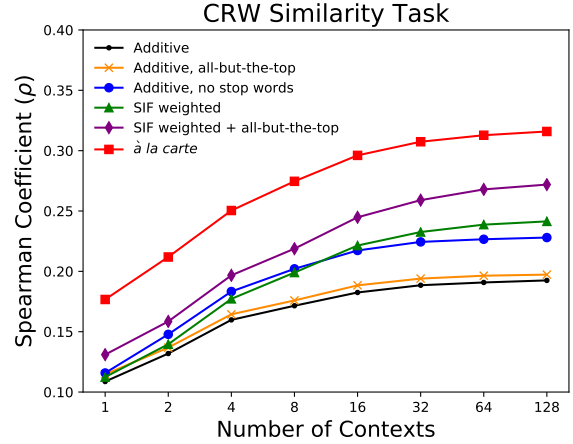


Figure 2: Spearman correlation between cosine similarity and human scores for pairs of words in the CRW dataset given an increasing number of contexts per rare word. Our *à la carte* method outperforms all previous approaches, even when restricted to only eight example contexts.

word2vec embeddings from scratch on the subcorpus together with the sampled contexts we achieve a Spearman correlation of 0.45; this gap between word2vec and our method shows that there remains room for even better approaches for few-shot learning of word embeddings.

4.2 Learning Embeddings of New Concepts: Nonces and Chimeras

We now evaluate our work directly on the tasks posed by Herbelot and Baroni (2017), who developed simple datasets and methods to “simulate the process by which a competent speaker encounters a new word in known contexts.” The general goal will be to construct embeddings of new concepts in the same semantic space as a known embedding vocabulary using contextual information consisting of definitions or example sentences.

Nonces: We first discuss the definitional nonce dataset made by the authors themselves, which has a test-set consisting of 300 single-word concepts and their definitions. The task of learning each concept’s embedding is simulated by removing or randomly re-initializing its vector and requiring the system to use the remaining embeddings and the definition to make a new vector that is close to the original. Because the embeddings were constructed using data that includes these concepts, an implicit assumption is made that including or excluding one word does not greatly affect the se-

Method	Nonce (Herbelot and Baroni, 2017)		Chimera (Lazaridou et al., 2017)		
	Mean Recip. Rank	Med. Rank	2 Sent.	4 Sent.	6 Sent.
word2vec	0.00007	111012	0.1459	0.2457	0.2498
additive	0.00945	3381	0.3627	0.3701	0.3595
additive, no stop words	0.03686	861	0.3376	0.3624	0.4080
nonce2vec	0.04907	623	0.3320	0.3668	0.3890
<i>à la carte</i>	0.07058	165.5	0.3634	0.3844	0.3941

Table 1: Comparison with baselines and nonce2vec (Herbelot and Baroni, 2017) on few-shot embedding tasks. Performance on the chimera task is measured using the Spearman correlation with human ratings. Note that the additive baseline requires removing stop-words in order to improve with more data.

mantic space; this assumption is necessary in order to have a good target vector for the system to be evaluated against.

Using 259,376 word2vec embeddings trained on Wikipedia as the base vectors, Herbelot and Baroni (2017) heavily modify the skip-gram algorithm to successfully learn on one definition, creating the *nonce2vec* system. The original skip-gram algorithm and $\mathbf{v}_w^{\text{additive}}$ are used as baselines, with performance measured as the mean reciprocal rank and median rank of the concept’s original vector among the nearest neighbors of the output.

To compare directly to their approach, we use their word2vec embeddings along with contexts from the Wikipedia corpus to construct context vectors \mathbf{u}_w for all words w apart from the 300 nonces. We then learn the *à la carte* transform \mathbf{A} , weighting the data points in the regression (4) using a hard threshold of at least 1000 occurrences in Wikipedia. An embedding for each nonce can then be constructed by multiplying \mathbf{A} by the sum over all word embeddings in the nonce’s definition. As can be seen in Table 1, this approach significantly improves over both baselines and nonce2vec; the median rank of 165.5 of the original embedding among the nearest neighbors of the nonce vector is very low considering the vocabulary size is more than 250,000, and is also significantly lower than that of all previous methods.

Chimeras: The second dataset Herbelot and Baroni (2017) consider is that of Lazaridou et al. (2017), who construct unseen concepts by combining two related words into a fake nonce word (the “chimera”) and provide two, four, or six example sentences for this nonce drawn from sentences containing one of the two component words. The desired nonce embeddings is then evaluated via the correlation of its cosine similar-

ity with the embeddings of several other words, with ratings provided by human judges.

We use the same approach as in the nonce task, except that the chimera embedding is the result of summing over multiple sentences. From Table 1 we see that, while our method is consistently better than both the additive baseline and nonce2vec, removing stop-words from the additive baseline leads to stronger performance for more sentences. Since the *à la carte* algorithm explicitly trains the transform to match the true word embedding rather than human similarity measures, it is perhaps not surprising that our approach is much more dominant on the definitional nonce task.

5 Building Feature Embeddings using Large Corpora

Having witnessed its success at representing unseen words, we now apply the *à la carte* method to two types of feature embeddings: synset embeddings and n -gram embeddings. Using these two examples we demonstrate the flexibility and adaptability of our approach when handling different corpora, base word embeddings, and downstream applications.

5.1 Supervised Synset Embeddings for Word-Sense Disambiguation

Embeddings of synsets, or sets of cognitive synonyms, and related entities such as senses and lexemes have been widely studied, often due to the desire to account for polysemy (Rothe and Schütze, 2015; Iacobacci et al., 2015). Such representations can be evaluated in several ways, including via their use for word-sense disambiguation (WSD), the task of determining a word’s sense from context. While current state-of-the-art methods often use powerful recurrent models (Raganato et al., 2017), we will instead use a sim-

Method	SemEval-2013 Task 12		SemEval-2015 Task 13			
	nouns	adj.	nouns	adv.	verbs	comb.
<i>à la carte</i> (SemCor)	60.0	72.2	67.7	85.2	60.6	68.1
<i>à la carte</i> (glosses)	51.8	75.3	62.5	79.0	55.8	64.2
<i>à la carte</i> (combined)	60.5	74.1	70.3	86.4	59.4	69.6
MFS (SemCor)	58.8	79.5	60.0	87.6	66.7	66.8
Raganato et al. (2017)	66.9					72.4

Table 2: Application of *à la carte* synset embeddings to two standard WSD tasks. As all systems always return exactly one answer, performance is measured in terms of accuracy. Results due to Raganato et al. (2017), who use a bi-LSTM for this task, are given as the recent state-of-the-art result.

ple similarity-based approach that heavily depends on the synset embedding itself and thus serves as a more useful indicator of representation quality. A major target for our simple systems is to beat the most-frequent sense (MFS) method, which returns for each word the sense that occurs most frequently in a corpus such as SemCor. This baseline is “notoriously hard-to-beat,” routinely besting many systems in SemEval WSD competitions (Navigli et al., 2013).

Synset Embeddings: We use SemCor (Langone et al., 2004), a subset of the Brown Corpus (BC) (Francis and Kucera, 1979) annotated using PWN synsets. However, because the corpus is quite small we use GloVe trained on Wikipedia instead of on BC itself. The transform \mathbf{A} is learned using context embeddings \mathbf{u}_w computed with windows of size ten around occurrences of w in BC and weighting each word by the log of its count during the regression stage (4). Then we set the context embedding \mathbf{u}_s of each synset s to be the average sum of word embeddings representation over all sentences in SemCor containing s . Finally, we apply the *à la carte* transform to get the synset embedding $\mathbf{v}_s = \mathbf{A}\mathbf{u}_s$.

Sense Disambiguation: To determine the sense of a word w given its context c , we convert c into a vector using the *à la carte* transform \mathbf{A} on the sum of its word embeddings and return the synset s of w whose embedding \mathbf{v}_s is most similar to this vector. We try two different synset embeddings: those induced from SemCor as above and those obtained by embedding a synset using its *gloss*, or PWN-provided definition, in the same way as a nonce in Section 4.2. We also consider a *combined* approach in which we fall back on the gloss vector if the synset does not appear in SemCor and thus has no induced embedding.

As shown in Table 2, synset embeddings induced from SemCor alone beat MFS overall, largely due to good noun results. The method improves further when combined with the gloss approach. While we do not match the state-of-the-art, our success in besting a difficult baseline using very little fine-tuning and exploiting none of the underlying graph structure suggests that the *à la carte* method can learn useful synset embeddings, even from relatively small data.

5.2 N-Gram Embeddings for Classification

As some of the simplest and most useful linguistic features, n -grams have long been a focus of embedding studies. Compositional approaches, such as sums and products of unigram vectors, are often used and work well on some evaluations, but are often order-insensitive or very high-dimensional (Mitchell and Lapata, 2010). Recent work by Poliak et al. (2017) works around this while staying compositional; however, as we will see their approach does not seem to capture a bigram’s meaning much better than the sum of its word vectors. n -grams embeddings have also gained interest for low-dimensional document representation schemes (Hill et al., 2016; Pagliardini et al., 2018; Arora et al., 2018a), largely due to the success of their sparse high-dimensional Bag-of- n -Grams (BonG) counterparts (Wang and Manning, 2012). This setting of document embeddings derived from n -gram features will be used for quantitative evaluation in this section.

We build n -gram embeddings using two corpora: 300-dimensional Wikipedia embeddings, which we evaluate qualitatively, and 1600-dimensional embeddings on the Amazon Product Corpus (McAuley et al., 2015), which we use for document classification. For both we use as source embeddings GloVe vectors trained on the respec-

Method	beef up	cutting edge	harry potter	tight lipped
$\mathbf{v}_{w_1} + \mathbf{v}_{w_2}$	meat, out	cut, edges	deathly, azkaban	loose, fitting
$\mathbf{v}_{(w_1, w_2)}^{\text{additive}}$	but, however	which, both	which, but	but, however
ECO	meats, meat	weft, edges	robards, keach	scaly, bristly
Sent2Vec	add, reallocate	science, multidisciplinary	naruto, pokemon	wintel, codebase
<i>à la carte</i>	need, improve	innovative, technology	deathly, hallows	worried, very

Table 3: Closest word embeddings (measured via cosine similarity) to the embeddings of four idiomatic or entity-associated bigrams. From these examples we see that purely compositional methods may struggle to construct context-aware bigram embeddings, even when the features are present in the corpus. On the other hand, adding up corpus contexts (1) is dominated by stop-word information. Sent2Vec is successful on half the examples, reflecting its focus on good sentence, not bigram, embeddings.

tive corpora over words occurring at least a hundred times. Context embeddings are constructed using a window of size ten and a hard threshold at 1000 occurrences is used as the word-weighting function in the regression (4). Unlike Poliak et al. (2017), who can construct arbitrary embeddings but need to train at least two sets of vectors of dimension at least $2d$ to do so, and Yin and Schutze (2014), who determine which n -grams to represent via corpus counts, our *à la carte* approach allows us to train exactly those embeddings that we need for downstream tasks. This, combined with our method’s efficiency, allows us to construct more than two million bigram embeddings and more than five million trigram embeddings, constrained only by their presence in the large source corpus.

Qualitative Evaluation: We first compare bigram embedding methods by picking some idiomatic and entity-related bigrams and examining the closest word vectors to their representations. These word-pairs are picked because we expect sophisticated feature embedding methods to encode a better vector than the sum of the two embeddings, which we use as a baseline. From Table 3 we see that embeddings based on corpora rather than composition are better able to embed these bigrams to be close to concepts that are semantically similar. On the other hand, as discussed in Section 3 and evident from these results, the additive context approach is liable to emphasize stop-word directions due to their high frequency.

Document Embedding: Our main application and quantitative evaluation of n -gram vectors is to use them to construct document embeddings. Given a length L document $D = \{w_1, \dots, w_L\}$, we define its embedding \mathbf{v}_D as a weighted con-

catenation over sums of our induced n -gram embeddings, i.e.

$$\mathbf{v}_D^T = \left(\sum_{t=1}^L \mathbf{v}_{w_t}^T \quad \dots \quad \frac{1}{n} \sum_{t=1}^{L-n+1} \mathbf{v}_{(w_t, \dots, w_{t+n-1})}^T \right)$$

where $\mathbf{v}_{(w_t, \dots, w_{t+n-1})}$ is the embedding of the n -gram (w_t, \dots, w_{t+n-1}) . Following Arora et al. (2018a), we weight each n -gram component by $\frac{1}{n}$ to reflect the fact that higher-order n -grams have lower quality embeddings because they occur less often in the source corpus. While we concatenate across unigram, bigram, and trigram embeddings to construct our text representations, separate experiments show that simply adding up the vectors of all features also yields a smaller but still substantial improvement over the unigram performance. The higher embedding dimension due to concatenation is in line with previous methods and can also be theoretically supported as yielding a less lossy compression of the n -gram information (Arora et al., 2018a).

In Table 4 we display the result of running cross-validated, ℓ_2 -regularized logistic regression on documents from MR movie reviews (Pang and Lee, 2005), CR customer reviews (Hu and Liu, 2004), SUBJ subjectivity dataset (Pang and Lee, 2004), MPQA opinion polarity subtask (Wiebe et al., 2005), TREC question classification (Li and Roth, 2002), SST sentiment classification (binary and fine-grained) (Socher et al., 2013), and IMDB movie reviews (Maas et al., 2011). The first four are evaluated using tenfold cross-validation, while the others have train-test splits.

Despite the simplicity of our embeddings (a concatenation over sums of *à la carte* n -gram vectors), we find that our results are very competitive with many recent unsupervised methods, achieving the best word-level results on two of the tested

Representation	n	d^*	MR	CR	SUBJ	MPQA	TREC	SST (± 1)	SST	IMDB
BonG	1	V_1	77.1	77.0	91.0	85.1	86.8	80.7	36.8	88.3
	2	$V_1 + V_2$	77.8	78.1	91.8	85.8	90.0	80.9	39.0	90.0
	3	$V_1 + V_2 + V_3$	77.8	78.3	91.4	85.6	89.8	80.1	42.3	89.8
<i>à la carte</i>	1	1600	79.8	81.3	92.6	87.4	85.6	84.1	46.7	89.0
	2	3200	81.3	83.7	93.5	87.6	89.0	85.8	47.8	90.3
	3	4800	81.8	84.3	93.8	87.6	89.0	86.7	48.1	90.9
Sent2Vec ¹	1-2	700	76.3	79.1	91.2	87.2	85.8	80.2	31.0	85.5
DisC ²	2-3	3200-4800	80.1	81.5	92.6	87.9	90.0	85.5	46.7	89.6
skip-thoughts ³		4800	80.3	83.8	94.2	88.9	93.0	85.1	45.8	
SDAE ⁴		2400	74.6	78.0	90.8	86.9	78.4			
CNN-LSTM ⁵		4800	77.8	82.0	93.6	89.4	92.6			
MC-QT ⁶		4800	82.4	86.0	94.8	90.2	92.4	87.6		
byte mLSTM ⁷		4096	86.8	90.6	94.7	88.8	90.4	91.7	54.6	92.2

* Vocabulary sizes (i.e. BonG dimensions) vary by task; usually 10K-100K.

^{1,3,7} (Pagliardini et al., 2018; Kiros et al., 2015; Radford et al., 2017) Evaluation conducted using latest pretrained models.

Note that the latest available skip-thoughts implementation returns an error on the IMDB task.

^{2,4,5,6} (Arora et al., 2018a; Hill et al., 2016; Gan et al., 2017; Logeswaran and Lee, 2018) Best results from publication.

Table 4: Performance of document embeddings built using *à la carte* n -gram vectors and recent unsupervised word-level approaches on classification tasks, with the character LSTM of (Radford et al., 2017) shown for comparison. Top three results are **bolded** and the best word-level performance is underlined.

datasets. The fact that we do especially well on the sentiment tasks indicates strong exploitation of the Amazon review corpus, which was also used by DisC, CNN-LSTM, and byte mLSTM. At the same time, the fact that our results are comparable to neural approaches indicates that local word-order may contain much of the information needed to do well on these tasks. On the other hand, separate experiments do not show a substantial improvement from our approach over unigram methods such as SIF (Arora et al., 2017) on sentence similarity tasks such as STS (Cer et al., 2017). This could reflect either noise in the n -gram embeddings themselves or the comparative lower importance of local word-order for textual similarity compared to classification.

6 Conclusion

We have introduced *à la carte* embedding, a simple method for representing semantic features using unsupervised context information. A natural and principled integration of recent ideas for composing word vectors, the approach achieves strong performance on several tasks and promises to be useful in many linguistic settings and to yield many further research directions. Of particular interest is the replacement of simple window contexts by other structures, such as dependency parses, that could yield results in domains such as question answering or semantic role labeling. Ex-

tensions of the mathematical formulation, such as the use of word weighting when building context vectors as in Arora et al. (2018b) or of spectral information along the lines of Mu and Viswanath (2018), are also worthy of further study.

More practically, the Contextual Rare Words (CRW) dataset we provide will support research on few-shot learning of word embeddings. Both in this area and for n -grams there is great scope for combining our approach with compositional approaches (Bojanowski et al., 2016; Poliak et al., 2017) that can handle settings such as zero-shot learning. More work is needed to understand the usefulness of our method for representing (potentially cross-lingual) entities such as synsets, whose embeddings have found use in enhancing WordNet and related knowledge bases (Camacho-Collados et al., 2016; Khodak et al., 2017). Finally, there remain many language features, such as named entities and morphological forms, whose representation by our method remains unexplored.

Acknowledgments

We thank Karthik Narasimhan and our three anonymous reviewers for helpful suggestions. The work in this paper was in part supported by SRC JUMP, Mozilla Research, NSF grants CCF-1302518 and CCF-1527371, Simons Investigator Award, Simons Collaboration Grant, and ONR-N00014-16-1-2329.

References

- Oliver Adams, Adam Makarucha, Graham Neubig, Steven Bird, and Trevor Cohn. 2017. Cross-lingual word embeddings for low-resource language modeling. In *Proc. EACL*.
- Sanjeev Arora, Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2018a. A compressed sensing view of unsupervised text embeddings, bag-of-ngrams, and lstms. In *Proc. ICLR*.
- Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. A latent variable model approach to pmi-based word embeddings. *TACL*.
- Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2018b. Linear algebraic structure of word senses, with applications to polysemy. *TACL*.
- Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *Proc. ICLR*.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *ArXiv*.
- Danushka Bollegala, David Weir, and John Carroll. 2014. Learning to predict distributions of words across domains. In *Proc. ACL*.
- José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *AI*.
- Daniel Cer, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and cross-lingual focused evaluation. In *Proc. SemEval*.
- Manaal Faruqui and Chris Dyer. 2014. Community evaluation and exchange of word vectors at word-vectors.org. In *Proc. ACL: System Demonstrations*.
- Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.
- W. Nelson Francis and Henry Kucera. 1979. *Brown Corpus Manual*. Brown University.
- Zhe Gan, Yunchen Pu, Ricardo Henao, Chunyuan Li, Xiaodong He, and Lawrence Carin. 2017. Learning generic sentence representations using convolutional neural networks. In *Proc. EMNLP*.
- Yoav Goldberg. 2016. A primer on neural network models for natural language processing. *JAIR*.
- Zellig Harris. 1954. Distributional structure. *Word*, 10:146–162.
- Aurélien Herbelot and Marco Baroni. 2017. High-risk learning: Acquiring new word vectors from tiny data. In *Proc. EMNLP*.
- Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *Proc. NAACL*.
- Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proc. KDD*.
- Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. Sensembed: Learning sense embeddings for word and relational similarity. In *Proc. ACL-IJCNLP*.
- Mikhail Khodak, Andrej Risteski, Christiane Fellbaum, and Sanjeev Arora. 2017. Automated wordnet construction using word embeddings. In *Proc. Workshop on Sense, Concept and Entity Representations and their Applications*.
- Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. In *Adv. NIPS*.
- Helen Langone, Benjamin R. Haskell, and George A. Miller. 2004. Annotating wordnet. In *Proc. Workshop on Frontiers in Corpus Annotation*.
- Angeliki Lazaridou, Marco Marelli, and Marco Baroni. 2017. Multimodal word meaning induction from minimal exposure to natural text. *Cognitive Science*.
- Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proc. COLING*.
- Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. In *Proc. ICLR*.
- Thang Luong, Richard Socher, and Christopher Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proc. CoNLL*.
- Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proc. ACL-HLT*.
- Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring networks of substitutable and complementary products. In *Proc. KDD*.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Adv. NIPS*.
- Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*.
- Andrea Moro and Roberto Navigli. 2015. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In *Proc. SemEval*.

- Jiaqi Mu and Pramod Viswanath. 2018. All-but-the-top: Simple and effective post-processing for word representations. In *Proc. ICLR*.
- Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. Semeval-2013 task 12: Multilingual word sense disambiguation. In *Proc. SemEval*.
- Sebastian Pado, Aurelie Herbelot, Max Kisselew, and Jan Snajder. 2016. Predictability of distributional semantics in derivational word formation. In *Proc. COLING*.
- Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proc. NAACL*.
- Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proc. ACL*.
- Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proc. ACL*.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proc. EMNLP*.
- Adam Poliak, Pushpendre Rastogia, M. Patrick Martin, and Benjamin Van Durme. 2017. Efficient, compositional, order-sensitive n-gram embeddings. In *Proc. EACL*.
- Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. ArXiv.
- Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2017. Neural sequence learning models for word sense disambiguation. In *Proc. EMNLP*.
- Sascha Rothe and Hinrich Schütze. 2015. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. In *Proc. ACL-IJCNLP*.
- Cyrus Shaoul and Chris Westbury. 2010. The westbury lab wikipedia corpus.
- Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. EMNLP*.
- Dingquan Wang, Nanyun Peng, and Kevin Duh. 2017. A multi-task learning approach to adapting bilingual word embeddings for cross-lingual named entity recognition. In *Proc. IJCNLP*.
- Sida Wang and Christopher D. Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proc. ACL*.
- Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Proc. LREC*.
- John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. In *Proc. ICLR*.
- Ledell Wu, Adam Fisch, Sumit Chopra, Keith Adams, Antoine Bordes, and Jason Weston. 2017. Starspace: Embed all the things! ArXiv.
- Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. In *Proc. CoNLL*.
- Wenpeng Yin and Hinrich Schutze. 2014. An exploration of embeddings for generalized phrases. In *Proc. ACL 2014 Student Research Workshop*.